Liberty Alliance Project: Single Sign -On

By Tan Sze Hou
GSEC Practical Assignment Version 1.4 Option 1
11 September 2002

Abstract

With the release of the Liberty Alliance Project Specification 1.0 [1] in July 15, 2002, a milestone has been reached for a n open standard Single Sign-On solution across multiple sites and services. Liberty is actually based on the Security Assertion Markup Language 1.0 (SAML) by OASIS [2]. Single Sign-On in Liberty refers to the ability to use proof of an existing authentica tion session with identit y provider to create a new authentication session with service providers. Single Sign-On is not a new concept but is available in many products currently on the market using their own propriet ary solution s.

Liberty makes used of existing web infrastructure for authentication and security and does not defined an y new methods for these. It uses XML to exchange proof of authentication and XML Signature [3] for signing message data to ensure data integrity and non -repudiation. However, there are some areas that version 1.0 of the specification does not addressed such as XML encryption and sharing of personal user information. This paper provides an overview of identity federation, Liberty architecture and the security issues that need to consider when evaluating future Liberty enabled product for Single Sign-On. In addition, this paper also discusse d some parts of the core protocol schema for liberty that may help the reader understand s the security issues.

What is Single Sign -On?

Most organizations today have Single Sign -On solutions to their portal or corporate web sites. This simplifies the administration of authentication and authorization through a central management server in the backend, which shares the information between app lications for the duration of the user session. In addition, user only needs to remember one password . This is an advantage as user can select s one strong password to remember instead of many simpl e passwords which are easily guess by using dictionary pass word lookup attack. However, all these solutions only work within their own corporate services and not across multiple sites and services . What about three or more parties such as a customer, merchant and a card issuer? [4]

Software that installed on the client side for password management also allows Single Sign -On by storing the password for each site in an encrypted database and makes them available to user upon login . User therefore only needs one password to access the database. The advantage is that you are using different password s for different site s but if your database is been compromise d all your pass words to each site will be reveal . One such popular product is v-GO and you can find more information by reading the GSEC practical “v-GO™ SSO: Sin gle Sign On or Single Password Solution? ” by Fredy Santoso. [5]

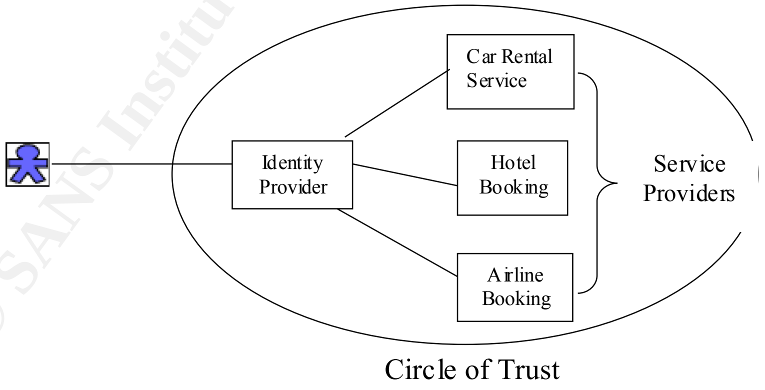
Liberty Alliance Project Single Sign-On solution brings a step further by creating federated identity between independent service providers. By allowing independent providers to use proof of an existing authentication session with identify provider to create a new authentication session with service providers. This allows two or more parties to particular in a Single Sign-On session. For example, a customer authenticated by the card issuer chooses to federate his/her identity to the merchant site. Later, he/she visit the merchant site and purchase products without re-authenticating with the card issuer. All these take place without exposing your personal information to the merchant site. [6]

Features of Liberty

As defined by version 1.0 of the specification, Liberty enabled the following features:
1) Opt-in account linking
2) Simplified sign-on for linked accounts
3) Authentication Context
4) Global log-out
5) Liberty alliance client feature
As we progress through the paper, features 1 to 3 will be highlight. This paper does not discussed on 4 and 5.

Federated Network Identity

Before we discussed on the identity federation let defined the following terms:
- Identity Provider – Creates, maintains, and manages identity information for Principals and provides Principal authentication to other service providers within a circle of trust.
- Service Provider – Provides services to Principals.



Federated Network Identity

Circle of trust (Federated Network Identity) is created through agreements between the identity provider and the service provider. Agreement definition or policy is out of scope for Liberty but something likes the certificate practice statement for Certificate Authority (CA) might be useful. Each principal may has more than one circle of trust for personal and corporate identification. Normally, service providers within a circle of trust have business relationship with each other. For example, when you go on a holiday oversea, you need airline, hotel to stay, and a car rental service to rent a car for driving around. Identity provider may be hosted by an independent third party or one of the service providers. The following illustrated a Single Sign-On with federated identity scenario:

- User after successful login at the airline booking web site is been presented with a list of service providers available in the circle of trust to federate his/her identity.
- In this case, the user chooses yes for the car r ental service.
- Later, the user visits the car rental service web site and is transparently logged in because he/she has established a federate identity with the car rental service.

Security Consideration

Liberty required the service provider to authenticate the identity of the identity provider through server-side X.509V3 certificate. The communication channel between both must be encrypted. Client authentication for the service provider is optional depending on the type of service but strongly recommended. By using Public Key Infrastructure (PKI), Liberty provides confidentiality (Channel only), data integrity, and non-repudiation. Confidentiality or privacy is simply keeping information secret, and is achieved through encrypted channel using asymmetric or symmetric key. Data integrity is the process of ensuring that the information is unchanged and non-repudiation means you cannot deny doing something. Both are achieved using digital signature at the message and channel levels. All these minimized the risk of spoofing of identity or service providers and man in the middle or session replay attack. Key management is one of the weakest links for PKI, and is not within the scope of Liberty. It is important for the proper management of private key and W3C is currently developing on XML Key Management Specification (XKMS).

In the case, where one of the service providers is compromised, the other service providers are not necessarily compromised too because Liberty does not transmitted principal information to the service providers, only proof (Known as Assertion – a piece of data regarding an act of authentication on the principal) that it is authenticated. In addition, service providers cannot communicate directly with the Principal but only through the identity provider. Of course, if the identity provider is been compromised, all the service providers will be affected. There are advantages and disadvantages to this architecture, but overall having to maintain and monitor one identity provider is much more efficient. This does not mean that for example, car rental service does not have their own authentication login without going through the identity providers.

Although Liberty federated identity is created to solve the problem of Single Sign-On across multiple web sites and services, it can be used within corporate portal or web site. By using PKI, Liberty can enhance the security of the corporate services against internal or external attack. Most corporate services including authorization/policy server are behind the DeMilitarized Zone (DMZ) and considered trust. Therefore, services or policy server do not authenticated each other identities. By using Liberty enabled policy server provides yet another layer of protection against hacker attack.

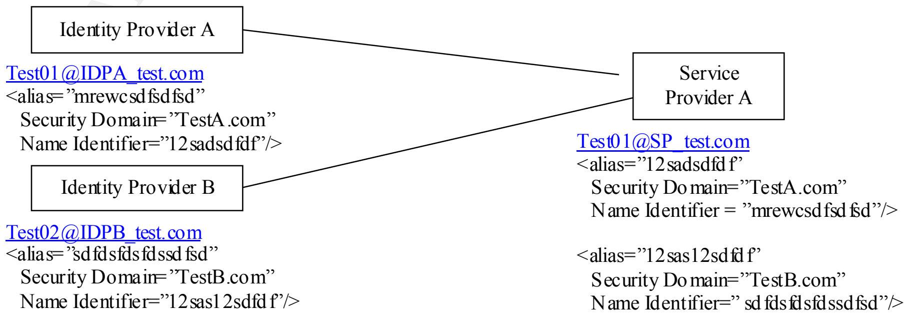
Next but not last is Liberty required Principal (user) approver to decide whether to federate or de-federate each service provider identities (One of the feature of Liberty Opt-in account linking is user can choose different service providers). Not only this gives the user a choice in selecting the required services but it prevent federate of identity to those services that are not required thereby minimizing security risk exposure. Furthermore, this protects the privacy of the user by refusing services that are push to them such as advertisement.

The identity provider is the security center for the circle of trust; therefore, it may be of great interest to the hacker. As the proof of authentication is in XML format, creating and processing a Document Object Model (DOM) object is both memory and CPU intensive jobs. Therefore, the identity provider is susceptible to denial of service attack by creating fake request. A possible buffer overflow may be exploited too if the implementation is not done properly.

There are other security considerations such as federation failures. For more information, please refer to the Liberty Specification [1].

Multiple Identity Providers

When there are more than one identity providers for the service provider, user needs to select one of the identity providers to login. How does the user know which identity providers are being logged in? Identity and service providers assigned a maximum of 256 pseudo-random characters name identifier for each login user. The identity provider name identifier must be global unique in order to solve the above problem (Name identifier needs not be global unique in the specification V1.0 but you have the problem of identifying two or more identity providers). Since the provider id for the identity provider is globally unique, we could take a MD5 hash of the provider id giving 32 characters (128bits). For each login user, the hash is appended as part of the name identifier. In this way, the service provider can uniquely identify more than one identity providers.



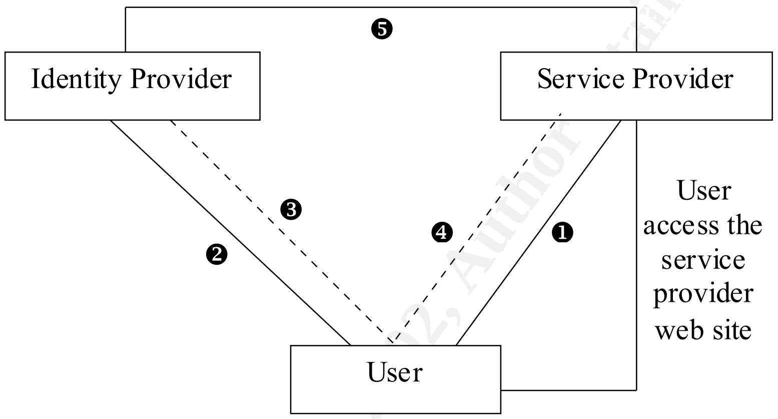
```
Test01@IDPA_test.com
<alias="mrewcsdfsdfsd"
  Security Domain="TestA.com"
  Name Identifier="12sadsdfdf"/>

Test02@IDPB_test.com
<alias="sdfdsfdsfdssdfsd"
  Security Domain="TestB.com"
  Name Identifier="12sas12sdfdf"/>

Test01@SP_test.com
<alias="12sadsdfdf"
  Security Domain="TestA.com"
  Name Identifier = "mrewcsdfsdfsd"/>

<alias="12sas12sdfdf"
  Security Domain="TestB.com"
  Name Identifier=" sdfdsfdsfdssdfsd"/>
```

A user with two identity providers federated to a service provider

We have looked at how service provider keeps track of multiple identity providers, so let looked at the security aspect. Name identifier generation including request id and artifact, which will cover in the next section, must not be guessable or in any related sequence from the previous values. Hijacking and replay attack is possible if the generated number can be predicted. To minimize the risks, name identifier is periodically refreshed but nothing is more important than a good opaque pseudo random generator.

Liberty Single Sign-On Protocol Flow

This section discussed web redirection between user, identity and service providers. This allows simplified sign-on for linked account.



Single Sign-On Protocol Flow

HTTP-Redirect

HTTP Redirect used the HTTP protocol to provide communication between the identity and service providers. First, when the user accesses the service provider web site or click on a link that required authentication for the services, the following protocol flow occurred:
❶ The service provider responses with a status code 302 (redirect) and an alternate URI in the header pointing to the identity provider. T he alternate URI embedded a URI with the name LRURL (recommended) pointing back to the service provider.
❷ The user agent does a redirect to the given identity provider URI. Not shown in the diagram above, the identity provider authenticated the user. Authentication can be any available technologies such as password, smart card, token, Kerberos, biometric or a combination of any of these.
❸ Whether, the authentication passed or failed, the identity provider response with a status code 302 and an alternate URI to the service provider pointing to itself. In addition, alternate URI embedded an artifact with the name SAMLart (required).
❹ The user agent does a redirect to the service provider URI.
❺ Using the artifact, the service provider sends a request (using SOAP protocol) to the identity provider to get the assertion. This is also known as the pull model (1 to 4 is known as the push model) . Based on the assertion, the

service provider either allowed or rejected the user to access the services or resources.

Form Based

Form Based method is almost similar to the HTTP Redirect but used the HTTP POST and scripting languages such as JavaScript or VBScript for redirection instead. The following protocol flow occurred:
❶ The service provider responses by retu rning a HTML page. Hidden fields containing the service provider URI and other additional information are embedded as parts of the form children in the HTML page. The form ac tion attribute is assigned to the URI of the identity p rovider and JavaScript or VBScript code in the page allowed auto submitting of the information by the user agent when the browser loaded the HTML page .
❷ Because of the script, the service provider URI and other information is auto posted to the identity provider.
Flow ❸ and ❹ are similar using the HTTP Post method to return the artifact back to the service provider.
Normally, form based method is used when service provide r has it own login page.

Cookies

Because cookies information cannot be shared across different domain (Identity and Service Providers most likely not be in the same domain), we cannot used cookies in this case unless we lower the security of the user agent.

Security Consideration

Although Liberty let the service provider decided whether to use HTTP, SSL/TLS or IPSEC for channel security , identity provider is required to do so . HTTP-Redirect and Form Based inheritance the security risks of today web based application. The next few paragraphs will discuss some of these issues .

As both methods depend on the use r agent for redirection, information may be cache by the user agent. This information may reveal confidential information about the authentication such as the artifact or password . Even worst, there are so many bugs reported for Microsoft Internet Explorer and hacker may exploits one of these to reveal your authentication information. Therefore, it is important that the user patch the system often.

Whether the authentication passed or failed , the identity provider returns an artifact back to the service p rovider. However, because artifact passed to the user agent for redirection, it is easily captured. Hacker could easily wr itten software capturing artifacts and looking for weakness in the artifact pseudo random generator implementation. Artifact is for on e time used but if we could predict the next string of characters generated (Like the TCP sequence number or session id), hacker can retrieve the assertion , modified it and passed to the service provider and accessed the services. Of course, it will not be so easy if we required the identity and service provider s to authenticate each other first. In addition, a weak or predictable artifact random character generator will weaken the two factor authentications that protect password stealing from Trojan or vir us program too. Therefore, one solution is to use two different algorithms for the generator, one for failure and the other for success. Hardware based random characters generator may be use for highly secure services.

Indeed interception of the communic ation flow is possible but the used of PKI for channel security minimized such risks. What PKI does not protected is poorly written program which allows the hacker to collect information about the system by looking at the HTML code and how information is p assed or even worst exploits bugs and weakness in the program such as through SQL Injection.

Protocol Schema

The bulk of Liberty schema is based on the Security Assertion Markup Language (SAML) specification [2]. It makes used of Simple Object Access Protocol (SOAP) binding and HTTP as the transport mechanism. Many elements make used of the timestamp encoded in UTC as one of the attributes. The purposes are to prevent replay attack and checking for the validity of the assertion .

Request and Response

Service provider issues an AuthnRequest and the identity provider responses with an AuthnResponse containing either the assertion and status or just the status indicating the failure. Both AuthnRequest and AuthnResponse have IssueInstant attribute timestamp .

Assertion

Assertion is a piece of data regarding an act of authentication on the principal. There are three types of assertion:

- Authentication – Specified a subject was authenticated by a particular means at a particular time (Used by SSO) .
- Authorization Decision - Allow the specified subject to access a specified resource. For example, whether to grant a request by subject for a n action on resource (such as web page or service) given the evidence.
- Attribute – The specified subject is associated with t he supplied attributes.

```
<saml:Assertion MajorVersion="1" MinorVersion="0" AssertionID="256.11.23.32.87654321"
Issuer="www.gsec.com " IssueInstant=" 2002-12-03T10:02:00Z ">
```

For each assertion, there is a unique AssertionID, Issuer (who issue the assertion) and IssuerInstant (timestamp) as an attributes of the Assertion element. It is important to synchronize the time between the identity and service providers. S o that additional check can be done by the identity or service provider to reject assertion that has wide variation in time from the creation to the time the assertion is retrieved.

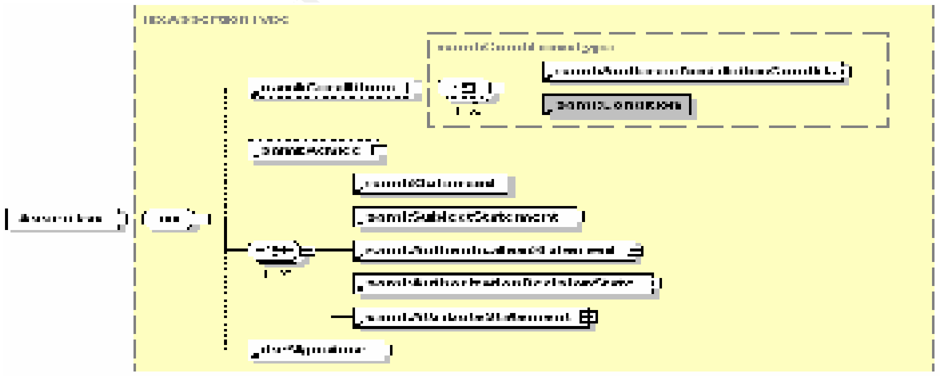
```
<saml:Conditions NotBefore="2002  -12-03T09:02:00Z"
NotOnOrAfter=" 2002-12-03T10:02:00Z " />
```

The saml:Conditions common element for each assertion types must be taken into account by the service provider. Which means service provider must rejected the assertion if the attribute or Conditions element is determined to be invalid or cannot be evaluated. The NotBefore (validity interval begins) and NotOnOrAfter (validity interval en ded) attributes of the saml:Conditions element determines the time range (timestamp) which the assertion is valid. As both attributes are optional, not specifying the NotOnOrAfter attribute means that the assertion is valid throughout. The same goes without specifying any attributes or the saml:Conditions element. It is a good practice to ensure that the time range be set to a minimum value allowed for the service to minimize the risk. Steps should be taken by the identity provider to check or remove the as sertion available if the validity period is over. The used of digital signature on the assertion ensures the data integrity of the conditions.

Another common element is the saml:Advice that contains any additional information that the issuer wishes to pr ovide. The service provider can ignore this.

```
<saml:Subject>
  <saml:NameIdentifier  Format="#emailAddress "/> gsec@test.com
  </saml:NameIdentifier>
     <saml:SubjectConfirmation>
        <saml:ConfirmationMethod>..URI..</saml:ConfirmationMethod>
     </saml:SubjectConfirmation>
</saml:Subject>
```

The saml:Subject element is a required field for all types of assertion. It identifies the subject by the name such as email address, X509 subject name or windows domain qualified name . In addition, the URI reference of the protocol used to authenticate the subject must be specified in the saml:ConfirmationMethod element .



Assertion Structure

saml:AuthenticationStatement:

```
<saml:AuthenticationStatement AuthenticationMethod="..URI.." AuthenticationInstant="    2002-
12-03T10:02:00Z">
  <saml:Subject>…</saml:Subject>
  <saml:SubjectLocality IPAddress="202.11.22.  20" DNSAddress=" 202.11.11.10 "/>
</saml.AuthenticationStatement>
```

An AuthenticationInstant attribute timestamp specifies the time which the authentication took place for the saml:AuthenticationStatement. This again allows the service provider to check for the variation and protect against replay attack. Another required attribute, AuthenticationMethod specify the type of authentication. The ip address and DNS address can be specify as an attribute of child element saml:SubjectLocality. Both of this any be easily spoof and should not be treated with confidence.

saml:AuthorizationDecisionStatement

```
<saml:AuthorizationDecisionStatement Decision=”Permit”
Resour ce=”http://www.gsec.com/test.html ”>
  <saml:Subject>…</saml:Subject>
  <saml:Action NameSpace=” urn:oasis:names:tc:SAML:1.0:action:rwedc  -negation”>Read
  </saml:Action>
</saml:AuthorizationDecisionStatemen  t>
```

The Decision attribute can be Permit, Deny or Indeterminate for the resource. The Evidence element, which is optional, can be added in to protect the issuer. This is something like a log of what the issuer relied on in issuing the authorization asserti on.

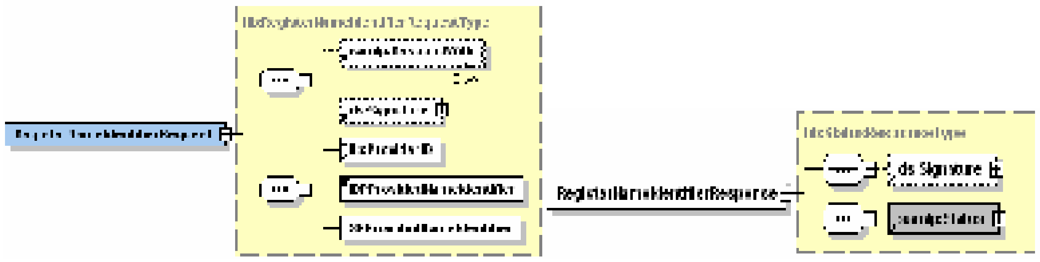
Saml:AttributeStatement

```
<saml:AttributeStatement>
  <saml:Subject>…</saml:Subject>
  <saml:Attribute AttributeName=”CreditLimit” AttributeNamespace=”http://smithco.com”>
    <saml:AttributeValue xsi:type=”my:type”>
      <my:amount currency=”U  SD”>500.00</my:amount>
    </saml:AttributeValue>
  </saml:Attribute>
</saml:AttributeStatement>
```

As shown in the example above, identity provider can provided useful information suc h as credit limit of the subject. This is useful in an authorization or transaction services. What missing is Liberty does not defined XML encryption. Encryption is useful for example , if you have a credit limit of 100000, you do not want people to know.

Name Identifier

As discussed in the previous section “multiple servi ce providers in the Federated Network Identity”, the service provider keeps track of the identity provider using name identifier. Service provider can refresh the name identifier after a certain time by sending a RegisterNameIdentifierRequest element putting the new name identifier in the SPProviderNameIdentifier as shown in the diagram.



Name Identifier Structure (Request and Response)

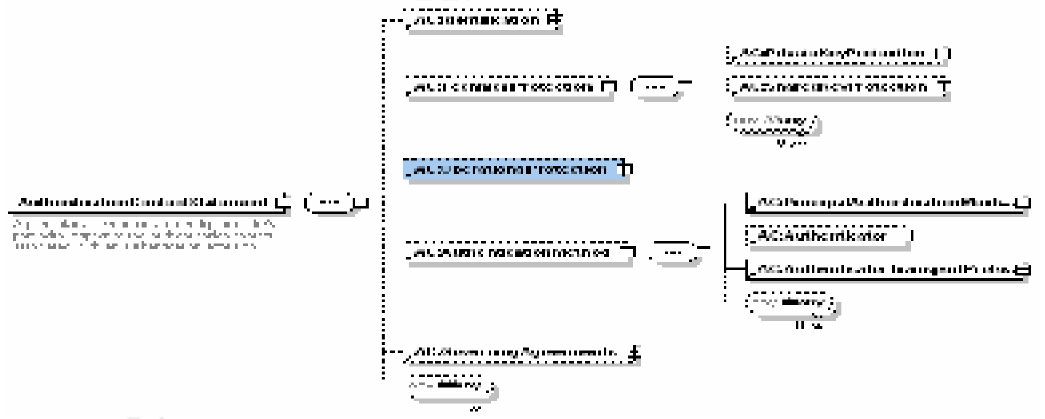
The identity provider responses with the RegisterNameIdentifierResponse element indicating whether it accept the chang es or not. A timestamp as an IssueInstant attribute on both the request and response elements protected again replay attack. As shown in the diagram digital signature, which is optional, can be used to provide data integrity for the request and response.

Digital Signature

As shown in the request and response, assertion, and name identifier in the previous section, Liberty makes used of XML digital signature (ds:Signature) for data integrity and non-repudiation (If public-private key pair). There are three types of XML signature, Enveloping, Enveloped, and Detached. Liberty is only using the enveloped XML signature type. Which means the signature is the child of an object being signed. Service and Identity providers are required to check the validity of the signature. Joseph Kee has written a good article on XML Signature, please read the article for more information [12].

Authentication Context

Authentication Context defined the information of how the authentication is carried out, user initial identification, protection of the private or share key and the operational protection and agreements by the identity provider. The Service provider based on above information, determines whethe r to place how much confidence on the assertion. Whether to include the authentication context (all or part of the information) as part of the assertion is optional and may not be provided by the identity provider.



Authentication Context Structure

Identification – This describes the process of how the principal (user) init ially register with the identity provider such as through on -line or face-to-face. Something likes the Certificate Practice Statement (CPS) on the registration of certificate with Certificate Authority (CA). For highly secure services such as banking, a face-to-face registration may be required.

Technical Protection – The process of how the private or share key is kept secure by the principal. For example, the storage media (smart card, token, Mobile Auth Card) and how secure is the key from been stolen. For password information such as the length, the minimum number of digit or ASCII characters to be a valid password. A good protection is impo rtant and increased confidence /validity of non-repudiation. Service provider providing services such as banking may reject the assertion because the password does not meet the minimum length required.

Operational Protection – Something similar to the PKI Disclosure Statement for CA that defined the security controls and audit employed. What security measure is in place to protect the aut hentication information , servers and security policy for emergency incident ? Physical security such as who has access to what information must be clearly list out. Overall, this determines how trust worth y the service provider sh ould trust the identity provider that the authentication information is not been stolen.

Authentication Method – This describes how the principal authenticated to the identity provider. Referring to the Authentication Context Structure, AuthenticatorT ransportProtocol element describes the channel security used in the authentication such as HTTP, SSL, MobileNetwork, IPSEC, and WTLS. PrincipalAuthenticationMechanism element captures the method of authentication such as token or password or smartcard . Lastly, the Authenticator element describes how the principal is authenticated. Overall, this determines whether the service provider will ask for a re-authentication, reject, or accept the assertion based on the type of services the principal is accessing.

Governing Agreements – A link for the agreement that defined business level authentication context such as liability and constraints between the service providers.

Conclusion

Liberty Alliance Project for Single Sig n-On is an open standard that protects consumer privacy and allows federated authentication from multiple service providers operating independently. This is accomplished by allowing exchange of assertion using XML. As shown in this paper, Liberty has considered many possible security issues an d offering flexibility in tracking these issues. However, a lot still depends on how each vendor implements the specification.

Although version 1.0 does not defined the exchange of user information according to permissions and preferences granted by user or XML Encryption for selected message confidential but depends on the channel sec urity such as SSL or TLS, nevertheless Liberty Alliance Project is an important step forward.

References

[1] Liberty Specification Version 1.0
URL: http://www.projectliberty.org/specs/liberty -specifications-v1.0.zip

[2] OASIS Committee Specifications. "SAML 1.0 Specification Set." May 31, 2002.
URL: http://www.oasis-open.org/committees/security/#documents

[3] W3C Recommendation. " XML-Signature Syntax and Processing ." February 12, 2002.
URL: http://www.w3.org/TR/xm ldsig-core/

[4] Laura Taylor. "Simplify, Save With Single Sign -On." May 29, 2002.
URL:
http://www.esecurityplanet.com/trends/article/0,,10751_1146321,00.html

[5] Fredy Santoso. "v-GO™ SSO: Single Sign On or Single Password Solution?" GSEC Practical. July 8, 2002.
URL: http://www.giac.org/practical/Fredy_Santoso_GSEC.doc

[6] Todd E. Sundsted. "With Liberty and Single Sign-On for all." February 15, 2002.
URL: http://www.javaworld.com/javaworld/jw -02-2002/jw-0215-liberty.html

[7] Edmund X. DeJesus. "SAML Brings Security to XML." February, 2002.
URL:
http://www.fawcette.com/xmlmag/2002_02/magazine/columns/collaboratio n/edejesus/default.asp

[8] Jon Byous. "SINGLE SIGN -ON SIMPLICITY WITH SAML: An Overview of Single Sign-On Capabilities Based on the Security Assertions Markup Language (SAML) Specification." May 9, 2002.
URL: http://java.sun.com/features/2 002/05/single-signon.html

[9] Eve Maler. "SAML basics – A Technical introduction to the Security Assertion Markup Language."
URL: http://www2002.org/presentations/maler.pdf

[10] James Kobielus . "SAML promises web services security." July 1, 2002.
URL: http://www.nwfusion.com/news/tech/2002/0701tech.html

[11] Tom Smith. "Liberty Alliance Specs Are Available; Seven Ven dors Detail Planned Products." July 15, 2002.
URL: http://www.internetwk.com/webDev/INW20020715S0008

[12] Joseph Kee. "XML Digital Signature by Example." GSEC Practical. May , 2002.
URL: http://www.giac.org/practical/Joseph_Kee_GSEC.doc